Bivariate Correlations and Descriptive Statistics

EOH710 Individual Project - Winter 2023

Lisa Frueh

2023-03-07

Read in Data

Shapefile was created in ArcGIS Pro by joining 2010 NY tracts with tract-level stressors (see Tract_Stressor_Sample.Rmd). Gal weights file was created in GeoDa.

```
ny <- st_read("./Data/shapefiles_weights/tract_stressor_sample.shp")
```

```
## Reading layer `tract_stressor_sample' from data source
##   `/Users/lisafrueh/Library/CloudStorage/OneDrive-DrexelUniversity/Research/EPA STAR/Spatial_Stresso
##   using driver `ESRI Shapefile'
## Simple feature collection with 4910 features and 47 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 105606.4 ymin: 4483095 xmax: 764133.3 ymax: 4985490
## Projected CRS: NAD83 / UTM zone 18N
```

```
#Read in weights file from GeoDa as an nb object--this creates a list of neighbors. In this case, first
queen.nb <- read.gal("./Data/shapefiles_weights/tract_ny_queen.gal", region.id=NULL, override.id=TRUE)

#Create spatial weights from neighbors list, style = "W" defines row-standardized weights-sums over all
lw <- nb2listw(queen.nb, glist=NULL, style="W", zero.policy=TRUE)
```

Define variables of interest

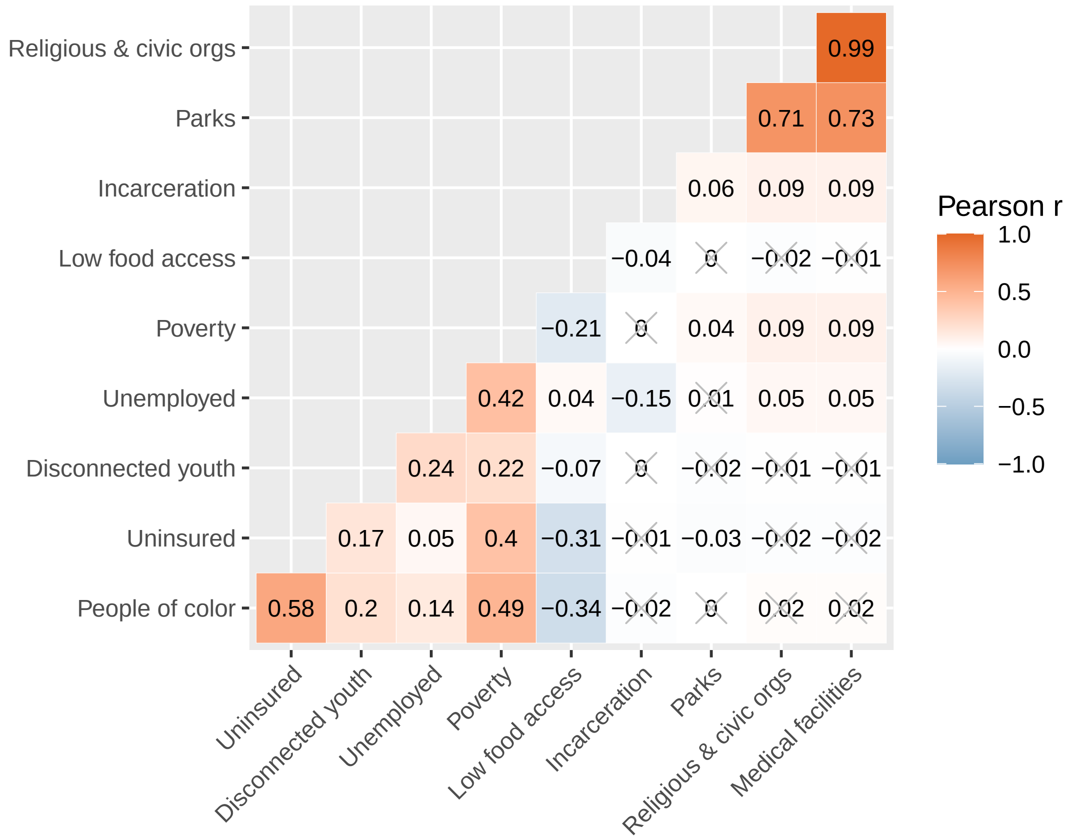
```
vars <-  c("park_a_per","la_05_10","unemp","fipr_100","poc","relciv_per","disc_youth","statepris1", "un

var_lab <- list(park_a_per = "Park area per 1,000 people", la_05_10 = "Low supermarket access", unemp =
```

Pearson rho correlation (non-spatial)

```r
#remove geometry
ny_ns <- st_drop_geometry(ny)

#calculate correlation matrix
rho.mat <- round(cor(ny_ns[vars]),2)

#store p values
p.mat <- cor_pmat(ny_ns[vars])

#Visualize
pearson_plot <-
ggcorrplot(rho.mat,
           p.mat = p.mat,
           hc.order=TRUE,
           type = "lower",
           outline.color="white",
           lab=TRUE,
           lab_size = 3,
           insig = "pch",
           pch.col = "gray",
           ggtheme = ggplot2::theme_gray,
           colors = c("#6D9EC1", "white", "#E46726"),
           legend.title = "Pearson r",
           tl.cex = 9
) +
  scale_x_discrete(labels = c("Uninsured", "Disconnected youth", "Unemployed", "Poverty", "Low food acce
  scale_y_discrete(labels = c("People of color", "Uninsured", "Disconnected youth", "Unemployed", "Pover

ggsave("pearson_plot.png", pearson_plot, device="png")
```

```
## Saving 6.5 x 4.5 in image
```

```r
pearson_plot
```



Lee's L statistic See: Lee (2001). Developing a bivariate spatial association measure: An integration of Pearson's r and Moran's I. J Geograph Syst 3: 369-385

In short, this method deduces a global (and local, but we'll just use global) statistic of bivariate correlation between two spatial variables.

```
#Check that the Global L statistic is symmetrical, like rho
poc_fipr100 <- lee(ny$poc, ny$fipr_100, lw, length(ny$poc), zero.policy=TRUE, NAOK=TRUE)
poc_fipr100$L
```

```
## [1] 0.3652213
```

```
#Monte-carlo simulation for bootstrapped p-value
test<- lee.mc(ny$poc, ny$fipr_100, nsim=99, lw, zero.policy=TRUE, alternative="two.sided", na.action=na

# It is symmetrical!
fipr100_poc <- lee(ny$fipr_100, ny$poc, lw, length(ny$fipr_100), zero.policy=TRUE, NAOK = TRUE)
fipr100_poc$L
```

```
## [1] 0.3652213
```

All bivariate pairs

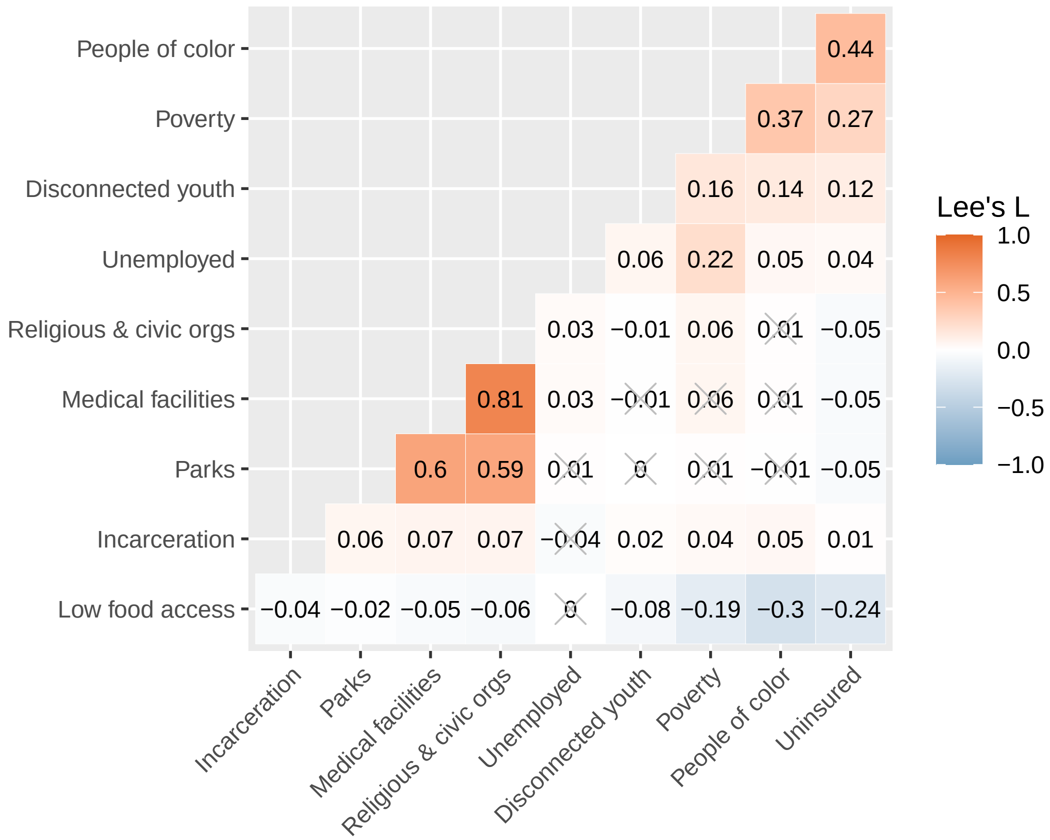
Write a function to run Lee's L-statistic for all combinations of variables specified in 'vars'.

```r
#Write a function that computes the L-statistic and p-value (via Monte Carlo simulation) for bivariate c
leepair <- function(d, x, y){
  test <- lee.mc(d[[x]], d[[y]], nsim=99, lw, zero.policy=TRUE, alternative="two.sided", na.action=na.om
  L <- test$statistic
  p <- test$p.value
  data.frame(var1=x, var2=y, L=L, p=p)
}

#Apply this function to all possible combinations of variables 1-11 in the list "vars"
#Then, combine these into one dataframe

df_total<-data.frame()
for(x in 1:10){
  for(y in 10:1){
    model <- plyr::ddply(ny, .(), leepair, x=vars[x], y=vars[y])
    df <- data.frame(model)
    df_total <- rbind(df_total,df)
  }
}

#Create a matrix to hold the L values and p values for bivariate combinations
lee.mat <- df_total %>%
  select(-.id, -p) %>%
  spread(var2, L) %>%
  data.frame(., row.names = .$var1) %>%
  select(-var1) %>%
  as.matrix(.)


lee.p.mat <- df_total %>%
  select(-.id, -L) %>%
  spread(var2, p) %>%
  data.frame(., row.names = .$var1) %>%
  select(-var1) %>%
  as.matrix(.)


#Visualize
L_plot <-
ggcorrplot(lee.mat,
           p.mat = lee.p.mat,
           hc.order=TRUE,
           type = "lower",
           outline.color="white",
           lab=TRUE,
           lab_size = 3,
           insig = "pch",
           pch.col = "gray",
           ggtheme = ggplot2::theme_gray,
           colors = c("#6D9EC1", "white", "#E46726"),
           legend.title = "Lee's L",
           tl.cex = 9
) +
```

```
  scale_x_discrete(labels = c("Incarceration", "Parks", "Medical facilities", "Religious & civic orgs",
  scale_y_discrete(labels = c("Low food access", "Incarceration", "Parks", "Medical facilities", "Relig

ggsave("L_plot.png", L_plot, device="png")
```

```
## Saving 6.5 x 4.5 in image
```

```
L_plot
```



Descriptive Statistics

```
ny_ns %>%
  select(all_of(vars)) %>%
  tbl_summary(
    label = var_lab,
    type = all_continuous() ~ "continuous2",
    statistic = all_continuous() ~ c(
      "{mean}",
      "{median} ({p25}, {p75})",
      "{min}, {max}"
    ),
    missing = "no"
```

```
  ) %>%
  bold_labels()
```

```
## Table printed with `knitr::kable()`, not {gt}. Learn why at
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in code chunk header.
```

Characteristic	**N = 4,910**
Park area per 1,000 people	
Mean	0.24
Median (IQR)	0.00 (0.00, 0.01)
Range	0.00, 262.11
Low supermarket access	
Mean	28
Median (IQR)	0 (0, 61)
Range	0, 100
% Unemployed	
Mean	37
Median (IQR)	36 (31, 41)
Range	0, 100
% Under poverty line	
Mean	16
Median (IQR)	12 (6, 22)
Range	0, 100
% People of color	
Mean	43
Median (IQR)	32 (12, 80)
Range	0, 100
Religious & civic organizations per 1,000 people	
Mean	4.12
Median (IQR)	1.54 (0.90, 2.46)
Range	0.00, 6,917.80
% Disconnected youth	
Mean	6
Median (IQR)	0 (0, 8)
Range	0, 100
# Incarcerated in state prisons, per 1,000 people	
Mean	499
Median (IQR)	116 (45, 311)
Range	0, 100,000
% Uninsured (18-64)	
Mean	13
Median (IQR)	12 (7, 18)
Range	0, 100
Ambulatory medical facilities per 1,000 people	
Mean	22.2
Median (IQR)	2.2 (0.9, 4.9)
Range	0.0, 77,556.1

```
ny_ns %>%
  select(all_of(vars), Urban) %>%
  mutate(urb = case_when(
    Urban==1 ~ "Urban",
    Urban==0 ~ "Rural",
    TRUE~NA_character_
  )) %>%
  tbl_summary(
    by = urb,
    label = var_lab,
    type = all_continuous() ~ "continuous2",
    statistic = all_continuous() ~ c(
      "{mean}",
      "{median} ({p25}, {p75})",
      "{min}, {max}"
    ),
    missing = "no"
  ) %>%
  modify_header(label ~ "**Variable**") %>%
  add_p(pvalue_fun = ~ style_pvalue(.x, digits = 2)) %>%
  bold_labels()
```

```
## Table printed with `knitr::kable()`, not {gt}. Learn why at
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in code chunk header.
```

Variable	**Rural**, N = 732	**Urban**, N = 4,178	**p-value**
Park area per 1,000 people			<0.001
Mean	0.67	0.16	
Median (IQR)	0.00 (0.00, 0.01)	0.00 (0.00, 0.01)	
Range	0.00, 148.46	0.00, 262.11	
Low supermarket access			<0.001
Mean	4	33	
Median (IQR)	0 (0, 0)	6 (0, 69)	
Range	0, 98	0, 100	
% Unemployed			0.025
Mean	36	37	
Median (IQR)	37 (33, 41)	36 (31, 41)	
Range	0, 100	0, 100	
% Under poverty line			<0.001
Mean	10	17	
Median (IQR)	10 (6, 14)	13 (6, 24)	
Range	0, 100	0, 100	
% People of color			<0.001
Mean	9	49	
Median (IQR)	5 (3, 9)	41 (18, 85)	
Range	0, 100	0, 100	
Religious & civic organizations per 1,000 people			0.006
Mean	1.94	4.50	
Median (IQR)	1.72 (1.12, 2.44)	1.50 (0.87, 2.46)	

Variable	**Rural**, N = 732	**Urban**, N = 4,178	p-value
Range	0.00, 51.07	0.00, 6,917.80	
% Disconnected youth			0.35
Mean	5	6	
Median (IQR)	1 (0, 6)	0 (0, 9)	
Range	0, 100	0, 100	
# Incarcerated in state prisons, per 1,000 people			<0.001
Mean	1,606	305	
Median (IQR)	88 (38, 148)	126 (46, 362)	
Range	0, 100,000	0, 37,500	
% Uninsured (18-64)			<0.001
Mean	10	14	
Median (IQR)	10 (6, 13)	12 (7, 18)	
Range	0, 100	0, 76	
Ambulatory medical facilities per 1,000 people			<0.001
Mean	1.8	25.8	
Median (IQR)	1.2 (0.5, 2.2)	2.5 (1.1, 5.5)	
Range	0.0, 28.5	0.0, 77,556.1	
Urban	0 (0%)	4,178 (100%)	<0.001